THE HARMFUL TAIL: TOWARD LOSSLESS QUANTIZATION OF LARGE LANGUAGE MODELS.

Anonymous authors
Paper under double-blind review

ABSTRACT

Low-bit quantization substantially reduces the cost of LLM inference, yet the relationship between error magnitude and predictive degradation remains poorly characterized. We show that this relationship is highly non-uniform across spectral directions. In W4A4-quantized Qwen3 models, dominant high-energy components of the quantization error contribute relatively little to performance loss whereas a low-energy spectral tail explains most of the excess loss. A gold aware analysis further reveals that correcting only four input-conditioned tail directions removes 0.43% of tail-error energy, while increasing mean downstream accuracy beyond the BF16 reference. We attribute this energy function mismatch to loss geometry. Tail directions exhibit systematically higher directional curvature per unit error energy. Motivated by this, we introduce Harmful-Tail-Guided Sparse Compensation, HTSC, which identifies modules whose local corrections propagate beneficially into harmful tail directions and distills this repair into sparse low-rank branches with a frozen backbone. At inference, the reference model and spectral decomposition are not required. Across Qwen3 models, HTSC restores performance to near or slightly above the BF16 reference, with low additional overhead. Code is available at: `https://anonymous.4open.science/r/HTSCptq-CE19/`.

1 INTRODUCTION

Low-bit quantization reduces the memory and computational costs of large language model (LLM) inference, but often degrades predictive accuracy. Existing methods commonly mitigate this degradation by reducing quantization error magnitude, either by minimizing discrepancies between quantized and floating-point outputs (Frantar et al., 2022) or by transforming weights and activations to improve quantizability (Xiao et al., 2023; Ashkboos et al., 2024). Although these approaches differ substantially in implementation, they share an implicit premise: larger numerical errors are generally more important to correct. Our results show that this premise can fail sharply.

We begin by examining the structure of the accumulated quantization error itself. At the final pre-normalization residual stream, where the effects of preceding quantized computations are presented directly to the final normalization and vocabulary readout, the error is strongly spectrally concentrated. A small number of singular directions capture a substantial fraction of the total error energy, forming a spectral head whose geometry is consistent across independently sampled documents.

To understand how this spectral structure arises, we trace quantization error from a local projection through subsequent Transformer blocks. Input statistics shape the local output-error spectrum and induce spectral concentration, while downstream propagation applies unequal gains to its components and maps distinct error directions into increasingly overlapping output subspaces. Together, input-conditioned error generation and downstream spectral reshaping produce a structured, concentrated terminal error spectrum.

This structured spectrum raises a more consequential question: are the dominant error directions also the directions most responsible for predictive degradation? Surprisingly, they are not.

We find a pronounced separation between error energy and predictive impact. Removing the spectral head of the final-residual quantization error increases negative log-likelihood (NLL), whereas removing its orthogonal tail recovers nearly all excess NLL and substantially restores the predictive distribution. The same disparity holds on downstream tasks: tail repair recovers accuracy close to

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

the floating-point reference, while head repair provides little improvement. Predictive damage is also highly sparse within the tail. A gold-aware oracle identifies just four input-conditioned directions whose correction removes only 0.43% of tail-error energy yet raises mean task accuracy above the floating-point reference.

The disparity in predictive impact between the head and tail calls for an explanation beyond error energy. We examine their interaction with the loss geometry of the final readout. First-order loss contributions cannot account for the advantage of tail repair: the tail has a more negative mean first-order contribution despite causing greater predictive degradation. Instead, tail errors exhibit higher directional curvature per unit energy, and second-order analysis closely predicts their greater repair benefit. The tail's larger quadratic loss contribution, rather than more harmful first-order alignment, explains its greater predictive impact under the evaluated readout.

These findings suggest a different objective for quantization correction. Rather than allocating correction capacity according to error magnitude, one should identify and repair the errors to which the model's predictive function is most sensitive. We propose **harmful-tail-guided sparse compensation** (HTSC) to transfer selective error correction into a deployable quantized model. HTSC first identifies modules whose local corrections propagate into harmful-tail directions and improve the final predictive distribution. It then use next-token supervision on domain text to construct an input-conditioned harmful-tail repair teacher and distill its predictive distribution into low-rank branches at the selected modules. The quantized backbone remains frozen, and inference requires only the learned sparse compensation branches.

We evaluate HTSC on Qwen3 models ranging from 0.6B to 30B-A3B, covering both dense and mixture-of-experts architectures, across eight downstream tasks and language-modeling perplexity on WikiText-2 and C4. Under W4A4 NVFP4 quantization, HTSC achieves a mean downstream accuracy of 71.86%, slightly exceeding the BF16 reference of 71.69%. It also maintains near-BF16 perplexity across the evaluated models. On Qwen3-8B, HTSC introduces only 0.13362% additional parameters . These results demonstrate that harmful-tail-guided sparse compensation can preserve full-precision predictive performance with small additional inference cost.

2 Spectral Structure and Formation of Quantization Error

2.1 Spectral Concentration and Cross-Document Consistency

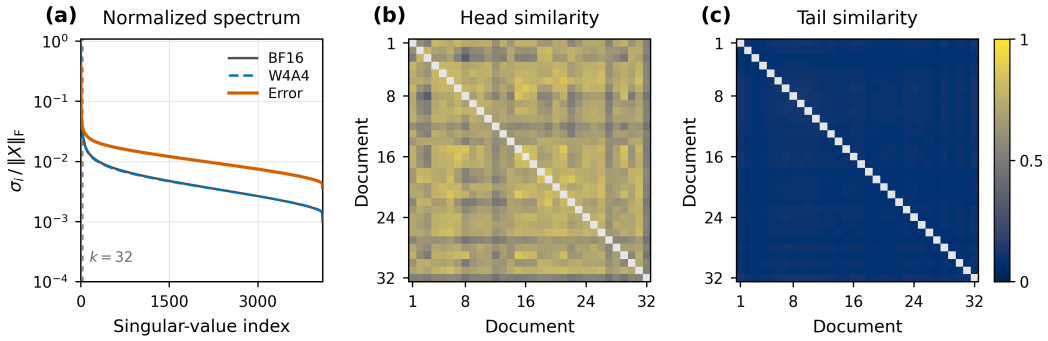

Figure 1: **Spectral concentration and cross-document consistency of final-residual quantization error.** (a) Frobenius-normalized singular-value spectra of the BF16 residual, W4A4 residual, and quantization error. (b,c) Pairwise Frobenius cosine similarity of the head and tail second-moment matrices across 32 DCLM documents (Li et al., 2024).

Quantization error concentrates its energy in a low-rank head. Let $H^{\mathrm{FP}}, H^{\mathrm{Q}} \in \mathbb{R}^{T\times d}$ denote the final pre-normalization residual states from paired BF16 and W4A4 forward passes of Qwen3-8B, where T is the sequence length and d is the hidden dimension. We construct the error matrix $E = H^{\mathrm{Q}} - H^{\mathrm{FP}}$ and perform uncentered SVD, $E = U\Sigma V^{\top}$. Its leading 32 singular directions, comprising only 0.78% of the hidden dimensions, capture 36.37% of the total error energy (Figure 1(a)), revealing substantial spectral concentration. Let $B \in \mathbb{R}^{d\times 32}$ contain the leading 32 right

singular vectors of E. We define the *error head* as $E_S = EBB^\top$ and its orthogonal complement, the *error tail*, as $E_T = E - E_S$.

The dominant error directions exhibit shared geometry across documents. For each document, we construct its error matrix $E^{(i)} \in \mathbb{R}^{T\times d}$ and independently apply the same uncentered rank-32 SVD to obtain its head $E_S^{(i)}$ and tail $E_T^{(i)}$, where i indexes the document. For each component $A \in \{S, T\}$, we define its uncentered second-moment matrix as $C_A^{(i)} = E_A^{(i)\top} E_A^{(i)}$ and compare documents using their Frobenius cosine similarity: $S_A(i,j) = \frac{\langle C_A^{(i)}, C_A^{(j)} \rangle_F}{\|C_A^{(i)}\|_F \|C_A^{(j)}\|_F}$, where $\langle \cdot, \cdot \rangle_F$ and $\|\cdot\|_F$ denote the Frobenius inner product and norm, respectively, and j indexes a second document. The head exhibits substantially higher cross-document similarity than the tail (Figure 1(b,c)), indicating that independently fitted low-rank error heads share a similar geometry in the hidden space.

2.2 Local Error Concentration and Downstream Spectral Reshaping

Input statistics concentrate local quantization error. To understand how the observed final-residual error structure arises, we examine how quantization error acquires its initial structure at a local source and how subsequent Transformer blocks reshape it. For input activations X collected over N positions and weight perturbation $\Delta W = Q(W) - W$, the weight-induced linear-output error $E_W = X\Delta W^\top$ has uncentered second moment $C_{\text{real}} = \frac{E_W^\top E_W}{N} = \Delta W C_X \Delta W^\top, C_X = \frac{X^\top X}{N}$. Under real DCLM inputs, the leading 32 directions capture 87.35% of the local output-error energy (Figure 2(a)). Thus, input statistics selectively weight parameter-quantization perturbations, producing a concentrated local output-error spectrum.

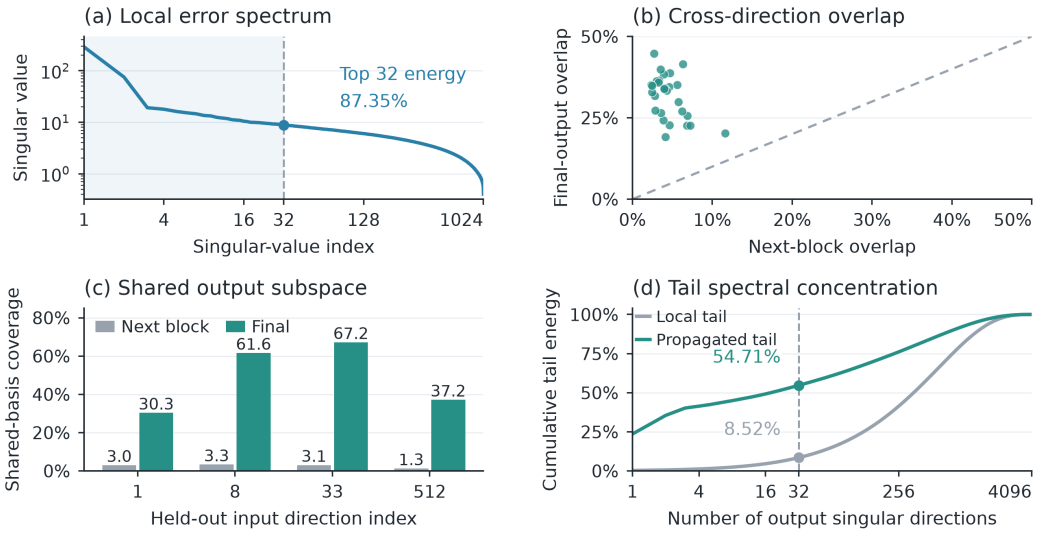

Figure 2: **Local error concentration and downstream spectral reshaping.** (a) Local output-error spectrum under real inputs. (b) Overlap of propagated error subspaces. (c) Shared-basis coverage of held-out error directions. (d) Spectral concentration of the local tail before and after propagation. Experiments use Qwen3-8B's block-7 gate projection and a differentiable FP32 surrogate for downstream propagation.

Downstream propagation selectively amplifies error components. The subsequent network transforms this locally structured error by applying different gains to its spectral components. We independently decompose the corresponding block-output residual perturbation D by uncentered rank-32 SVD into its head D_S and orthogonal tail D_T. Under the full-sequence downstream Jacobian $\mathcal{J}$, we measure the energy gain $G_A^2 = \|\mathcal{J}[D_A]\|_F^2 / \|D_A\|_F^2$ for $A \in \{S, T\}$. Across DCLM documents, the local tail receives 6.6–84.2 times the final-residual energy gain of the local head. Downstream propagation therefore changes the relative strength of the local error components, while their resulting spectral structure also depends on how these components are mapped into the output space.

Shared output directions concentrate the propagated tail. Downstream propagation further reshapes the local error by mapping distinct components into partially shared dominant output subspaces. We propagate unit-norm singular components $u_j v_j^\top$ of D to isolate output geometry from their initial magnitudes. Their dominant output hidden subspaces become more overlapping toward the final residual (Figure 2(b)), while a shared rank-32 basis captures substantially more energy from held-out components at the final residual than at the next block (Figure 2(c)). Correspondingly, the leading 32 singular directions of the complete local tail capture 8.52% of its energy before propagation and 54.71% afterward (Figure 2(d)). These results show that input-conditioned local concentration and downstream spectral reshaping provide complementary mechanisms through which structured quantization error can emerge in the final residual.

3 Energy–Function Separation in Quantization Error

3.1 Disproportionate Functional Impact of the Spectral Tail

Error energy does not determine predictive impact. We examine whether the spectral energy of final-residual quantization error reflects its contribution to predictive degradation. We independently remove document-specific head E_S and tail E_T from the quantized residual, obtaining $H^{(-S)} = H^{\mathrm{Q}} - E_S$ and $H^{(-T)} = H^{\mathrm{Q}} - E_T$. We compare their NLL recovery with full-residual restoration, $H^{(-E)} = H^{\mathrm{FP}}$. As shown in Figure 3(a), removing the head increases NLL, whereas tail repair recovers essentially all the excess NLL. The functional impact of quantization error is therefore strongly dependent on its spectral directions rather than its energy alone.

Tail repair also restores the predictive distribution. We further measure the vocabulary-level KL divergence from the BF16 predictive distribution under the same interventions. Tail repair substantially reduces this divergence, while head repair yields only a modest improvement (Figure 3(b)).

Shared-basis tail repair recovers downstream accuracy. Motivated by the shared geometry of dominant error directions across documents, we evaluate head and tail repair using a shared rank-32 basis on five zero-shot tasks. Tail repair restores average accuracy close to BF16, whereas head repair provides little improvement over W4A4. This advantage persists under controlled comparisons: repairing the top 32 tail directions outperforms the 32-dimensional head repair, while repairing tail directions that collectively account for 37.7% of the total error energy reaches 69.43% (Table 1).

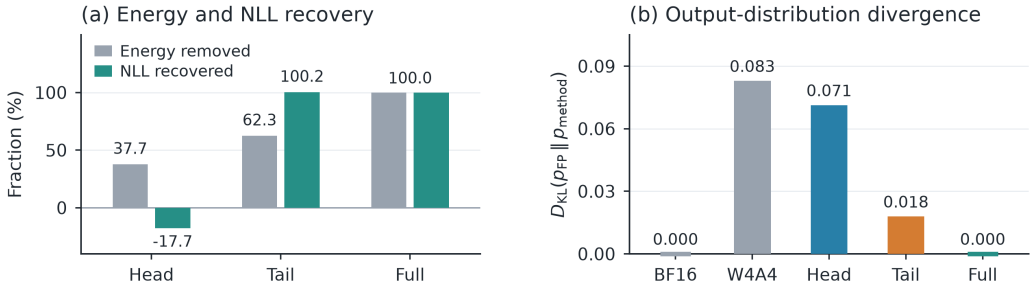

Figure 3: **Predictive impact of spectral error components.** (a) Error energy removed and excess-NLL recovery under head, tail, and full-residual repair. (b) Vocabulary-level KL divergence from the BF16 predictive distribution under the same interventions.

Table 1: **Zero-shot downstream accuracy (%).** Qwen3-8B with W4A4 quantization. The controlled tail repairs use task-specific bases obtained by uncentered rank-32 SVD.

Condition	ARC-C	ARC-E	HellaSwag	PIQA	RACE	Avg.
BF16	**56.74**	**80.98**	74.91	**77.69**	**58.94**	**69.85**
W4A4	53.67	79.92	73.34	75.24	57.76	67.99
Head repair	52.56	80.01	73.70	77.04	57.82	68.23
Tail repair	56.66	80.85	**75.06**	77.53	58.09	69.64
Tail top-32 repair	55.97	79.55	72.93	76.82	58.15	68.68
Tail 37.7%-energy repair	56.31	80.43	74.91	77.20	58.31	69.43

3.2 Loss Geometry of Spectral-Head and Spectral-Tail Errors

First-order alignment cannot explain the repair advantage. We analyze the loss geometry of document-specific rank-32 final-residual errors using Qwen3-8B under W4A4 quantization, 192 DCLM documents, and a fixed differentiable readout. For each component $A \in \{S, T\}$, we measure its signed first-order contribution to mean next-token NLL L as $a_A = \langle \nabla L(H^{\mathrm{FP}}), E_A \rangle_F$. Both contributions are negative for most documents, with a more negative mean for the tail (Figure 4(a)). Thus, first-order alignment cannot explain why tail repair yields greater NLL recovery.

The spectral tail has higher directional curvature. We next measure the curvature of each error component using the exact token-loss Hessian $\mathcal{H}_t$ at the FP residual h_t^{FP}. For $A \in \{S, T\}$, we define curvature per unit error energy as

$$\rho_A^{\mathcal{H}} = \frac{\sum_t e_{A,t}^\top \mathcal{H}_t e_{A,t}}{\sum_t \|e_{A,t}\|_2^2}, \tag{1}$$

where $e_{A,t}$ is the error component at token t. The tail has higher curvature density in all 192 documents, with a median tail-to-head ratio of 2.35 (Figure 4(b)). For the generalized Gauss–Newton matrix G of the readout, $e^\top G e = \mathrm{Var}_{v\sim p}[(J_z e)_v]$, where J_z is the residual-to-logit Jacobian and p is the FP predictive distribution. Thus, directional sensitivity depends on the induced variation in logits rather than residual-space energy alone. The exact Hessian also includes contributions from readout nonlinearity.

Second-order geometry predicts the repair advantage. We test whether the curvature difference explains the measured benefit of repairing each component. Let $q_A = \frac{1}{2N} \sum_t e_{A,t}^\top \mathcal{H}_t e_{A,t}$ denote its quadratic loss contribution, where N is the number of valid prediction positions, and let $R_A = L(H^{\mathrm{Q}}) - L(H^{\mathrm{Q}} - E_A)$ denote its repair benefit. The tail-over-head repair advantage is $\Delta R \equiv R_T - R_S = L(H^{\mathrm{FP}} + E_T) - L(H^{\mathrm{FP}} + E_S)$, with second-order approximation $\widehat{\Delta R}^{(2)} = (a_T - a_S) + (q_T - q_S)$. The second-order prediction closely matches the measured repair advantage across documents, reducing prediction error by more than an order of magnitude relative to the first-order prediction (Figure 4(c)). The second-order approximations also capture the main separation between finite head- and tail-repair trajectories, although they do not fully explain effects such as the benefit of partial head repair (Figure 4(d)). Under the evaluated readout, the tail's larger quadratic loss contribution, rather than first-order alignment, explains its greater repair benefit.

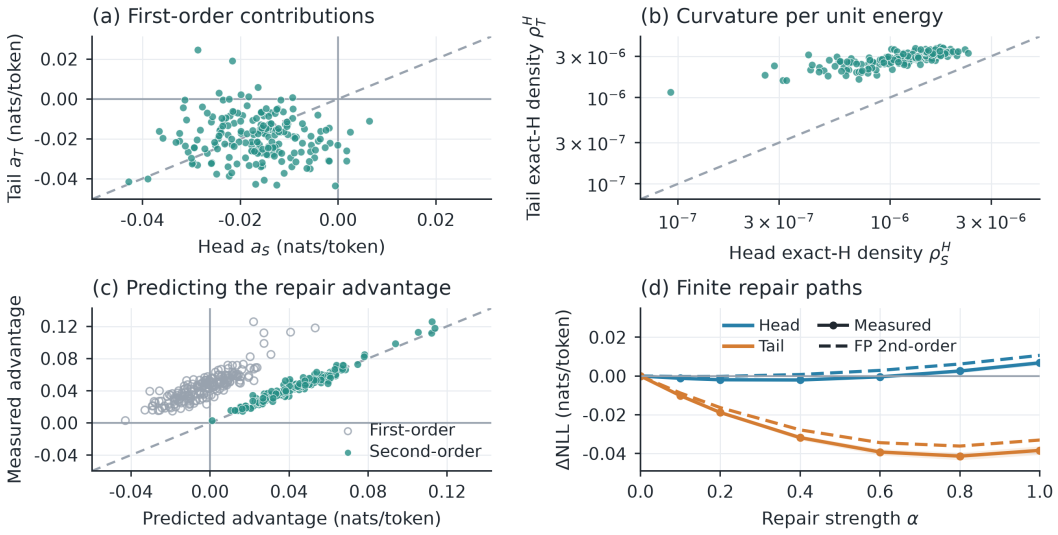

Figure 4: **Loss geometry of spectral-head and spectral-tail errors.** Qwen3-8B results on 192 DCLM documents. (a) Signed first-order loss contributions. (b) Exact-Hessian curvature per unit error energy. (c) Predicted versus measured tail-over-head repair advantage. (d) Measured and second-order-predicted NLL changes along repair paths. Shaded bands indicate 95% bootstrap confidence intervals.

3.3 Sparse Task-Harmful Directions in the Quantization Error Tail

Task degradation is concentrated in a few tail directions. We use a gold-aware oracle to examine whether repairing a few task-harmful directions can recover performance without correcting the entire tail. For each task, we use its cross-fitted rank-32 head basis B from Section 3.1 and construct an orthonormal basis $V_T = [v_1, \ldots, v_{d-32}]$ for the complementary tail subspace. For each held-out question, let J denote the correct-choice log probability obtained by applying a temperature-scaled softmax ($\tau = 0.05$) to length-normalized choice log-likelihoods. With paired residual error $e_{c,t} = h^{\mathrm{Q}}_{c,t} - h^{\mathrm{FP}}_{c,t}$ and objective gradient $g_{c,t} = \nabla_{h^{\mathrm{Q}}_{c,t}} J$ for choice c at token t, the first-order benefit of correcting direction v_j is

$$b_j = -\sum_{c,t} (e_{c,t}^\top v_j)(g_{c,t}^\top v_j). \tag{2}$$

We remove the error projections along up to k directions with the largest positive b_j from all answer choices. Because direction selection requires the gold label and paired BF16–W4A4 residuals, this oracle measures the potential of selective repair rather than a deployable inference procedure.

Sparse tail correction yields disproportionate performance gains. Correcting just four directions removes only 0.43% of tail-error energy but raises mean accuracy from 67.99% under W4A4 to 71.37%, exceeding the BF16 reference of 69.85% (Table 2). Increasing k further improves accuracy while removing only a small fraction of tail-error energy. These results show that task-harmful error is concentrated in a small, input-conditioned subset of the tail.

Table 2: **Sparse task-harmful tail repair.** Zero-shot accuracy (%) under gold-aware correction of up to k directions using task-specific, four-fold cross-fitted rank-32 head bases. Energy is the mean percentage of tail-error energy removed across five tasks.

Condition	k	Energy (%)	ARC-C	ARC-E	HellaSwag	PIQA	RACE	Avg.
BF16	–	–	56.74	80.98	74.91	77.69	58.94	69.85
W4A4	–	–	53.67	79.92	73.34	75.24	57.76	67.99
Harmful-tail	4	0.43	59.22	82.24	74.84	78.94	61.63	71.37
	8	0.77	61.43	84.39	76.11	80.41	63.96	73.26
	16	1.38	65.27	86.49	78.07	83.46	67.69	76.20
	32	2.40	71.33	89.90	80.97	87.60	72.58	80.48

4 Harmful-Tail-Guided Sparse Compensation

Correcting a few harmful-tail directions can improve predictions, but the oracle requires a high-precision reference and supervision unavailable at inference. We transfer selective repair into a deployable quantized model by constructing an input-conditioned harmful-tail teacher, identifying modules whose corrections can approach its predictive distribution, and distilling the teacher into sparse low-rank compensation branches. The quantized backbone remains frozen.

4.1 Input-Conditioned Harmful-Tail Teacher

Construct selective repair targets from next-token supervision. We construct the repair teacher using observed next tokens in domain training text. For each training document, we obtain paired BF16 and quantized final residuals and define their error as $E = H^{\mathrm{Q}} - H^{\mathrm{FP}}$. We fit an uncentered error basis using four-fold document-level cross-fitting, excluding each document from the data used to estimate its basis. The leading 32 directions define the structured head, and the remaining right singular vectors form an orthonormal basis of its complementary tail.

For each document x, let $e_t = h^{\mathrm{Q}}_t - h^{\mathrm{FP}}_t$ denote its residual error at token t, and let g_t be the gradient of mean next-token NLL with respect to h^{Q}_t. For a direction v_j in the document's cross-fitted tail basis, its exact error correction is $c_{t,j} = -v_j v_j^\top e_t$, with first-order predicted NLL reduction $b_j = -\sum_t g_t^\top c_{t,j}$. We select up to 32 directions with the largest positive b_j, using a common

direction set $\mathcal{I}(x)$ across the document. Let $V_H(x)$ contain the selected directions and $P_H(x) = V_H(x)V_H(x)^\top$ denote their orthogonal projector. The selectively repaired teacher residual is $h_t^T = h_t^{\mathrm{Q}} + \sum_{j\in\mathcal{I}(x)} c_{t,j}$, or equivalently $H^T = H^{\mathrm{Q}} - (H^{\mathrm{Q}} - H^{\mathrm{FP}})P_H(x)$ in matrix form. The frozen downstream mapping $\mathcal{R}$ maps h_t^T to the teacher distribution $p_T(\cdot \mid x_{\le t})$.

The same training documents, cross-fitted bases, and document-specific repair targets are used for module selection and subsequent distillation. Teacher targets are generated from the uncompensated quantized model and remain fixed during compensation training.

4.2 Harmful-Tail-Guided Module Selection

Select modules by their propagated benefit to harmful directions. Since downstream propagation reshapes local errors, we evaluate candidate modules by whether their corrections move final-residual predictions toward the harmful-tail teacher, rather than by local quantization-error magnitude. For a candidate correction producing final residual H, we retain only its propagated change along the teacher's selected directions, $\widetilde{H} = H^{\mathrm{Q}} + (H - H^{\mathrm{Q}})P_H(x)$. We measure its agreement with the teacher using

$$\mathcal{L}_{\mathrm{sel}}(x; H) = \frac{1}{T}\sum_t D_{\mathrm{KL}}\left(p_T(\cdot \mid x_{\le t}) \,\middle\|\, \mathcal{R}(\widetilde{h}_t)\right), \tag{3}$$

where T is the number of valid prediction positions. For each document, the teacher distribution and projector $P_H(x)$ remain fixed when comparing candidate modules. This objective evaluates the predictive effect of each module's correction after downstream propagation, restricted to the harmful-tail directions identified by the teacher.

Screen and verify local correction sites. For each candidate linear module m, we use its weight-quantization residual $r_{m,t} = (W_m - Q(W_m))x_{m,t}$ as a local correction probe, where $x_{m,t}$ is its quantized input. We estimate the probe's benefit to the selection objective using

$$S_m^{\mathrm{grad}} = -\mathbb{E}_x \sum_t \left\langle \frac{\partial \mathcal{L}_{\mathrm{sel}}}{\partial y_{m,t}}, r_{m,t} \right\rangle, \tag{4}$$

evaluated at the uncompensated quantized model. We screen modules using this score, then verify promising candidates by injecting a finite correction $y_m^{\mathrm{Q}} + \epsilon r_m$ and measuring the resulting reduction in $\mathcal{L}_{\mathrm{sel}}$ through a complete forward pass. We select up to 14 modules with positive verified benefit under the prescribed selection budget. The selected set $\mathcal{S}$ is fixed before compensation training.

4.3 Sparse Low-Rank Compensation and Distillation

Distill selective repair into trainable module corrections. For each selected module $m \in \mathcal{S}$, we add a rank-r branch with $r = 64$ to its quantized output:

$$y_m = y_m^{\mathrm{Q}} + B_m A_m x_m, \qquad A_m \in \mathbb{R}^{r\times d_{\mathrm{in},m}}, \quad B_m \in \mathbb{R}^{d_{\mathrm{out},m}\times r}. \tag{5}$$

We initialize B_m to zero and train only the compensation branches while keeping the quantized backbone and final readout frozen. Using the same domain training documents and fixed teacher distributions as in module selection, the student minimizes token-level distributional distillation:

$$\mathcal{L}_{\mathrm{KD}} = \frac{1}{T}\sum_t D_{\mathrm{KL}}\left(p_T(\cdot \mid x_{\le t}) \,\|\, p_\theta(\cdot \mid x_{\le t})\right). \tag{6}$$

Unlike the projected candidate corrections used for module selection, the student branches are not constrained to change only the final-residual tail. Instead, they learn the teacher's predictive correction through distillation. We select the training checkpoint using held-out domain-text NLL.

4.4 Inference and Compensation Overhead

Inference requires only sparse low-rank branches. At inference, only the quantized backbone and learned branches remain; the BF16 reference, teacher, spectral bases, and direction-selection procedure are discarded. The additional parameter count is $P_{\mathrm{add}} = \sum_{m\in\mathcal{S}} r(d_{\mathrm{in},m} + d_{\mathrm{out},m})$. Each executed branch adds two low-rank linear operations without changing the attention structure or KV-cache dimensions.

5 EXPERIMENTS

5.1 EXPERIMENTAL SETUP

Models and benchmarks. We evaluate Qwen3-0.6B, Qwen3-4B, Qwen3-8B, and Qwen3-30B-A3B, covering dense and mixture-of-experts architectures. We report accuracy on eight downstream tasks: ARC-C, ARC-E, GSM8K, HellaSwag, LAMBADA, MMLU, PIQA, and WinoGrande. Downstream evaluation uses the LM Evaluation Harness (Gao et al., 2024). We also evaluate language-modeling perplexity on WikiText-2 (Merity et al., 2016) and C4 (Raffel et al., 2020). Average downstream accuracy is the unweighted mean across the eight tasks.

Baselines and implementation. We compare HTSC against the BF16 reference, the direct quantization baseline NVFP4-RTN, and four quantization-improvement methods: SmoothQuant (Xiao et al., 2023), SpinQuant (Liu et al., 2025), FlatQuant (Sun et al., 2025), and OCGQuant (Yao et al., 2026). These methods cover activation–weight smoothing and calibration- or transformation-based quantization approaches. All quantized variants follow a shared W4A4 NVFP4 fake-quantization protocol (NVIDIA, 2026). Calibration-based methods and HTSC use the same 128 sequences of 2,048 tokens sampled from the WikiText-2 training split with seed 0. HTSC constructs an input-conditioned harmful-tail teacher and trains sparse rank-64 compensation branches while keeping the quantized backbone frozen. Quantization, calibration, task evaluation, and perplexity computation are detailed in Appendix A, while HTSC-specific training details, including teacher preparation, module selection, optimization, and checkpoint selection, are provided in Appendix B.

5.2 MAIN RESULTS

Downstream performance recovery. Table 3 reports downstream accuracy on Qwen3-8B and Qwen3-30B-A3B. Under W4A4 NVFP4 quantization, HTSC recovers the accuracy lost to quantization, achieving a mean accuracy of 71.86%, exceeding the BF16 reference by 0.17 percentage points, with only 0.13362% additional parameters. Results on Qwen3-0.6B and Qwen3-4B are provided in Appendix C, while WikiText-2 and C4 perplexity results are reported in Appendix D.

Table 3: **Downstream accuracy (%).** Results on Qwen3-8B and Qwen3-30B-A3B under W4A4 NVFP4 quantization. Avg. is the unweighted mean across eight tasks. Results for Qwen3-0.6B and Qwen3-4B are provided in Appendix C. Higher is better.

Model	Method	ARC-C	ARC-E	GSM8K	HellaSwag	LAMBADA	MMLU	PIQA	WinoG.	Avg.
Qwen3-8B	BF16	56.74	80.98	76.72	74.91	**64.25**	**74.98**	77.69	67.25	71.69
	NVFP4-RTN	53.67	79.92	70.43	73.34	61.71	72.59	75.24	**69.45**	69.55
	SmoothQuant	54.52	80.14	78.01	72.89	63.92	72.97	75.90	66.14	70.56
	SpinQuant	52.05	75.93	71.04	72.64	63.89	70.29	76.22	65.19	68.41
	FlatQuant	54.35	78.45	77.10	72.75	62.10	71.79	75.46	66.38	69.80
	OCGQuant	54.61	80.26	**81.05**	73.55	64.10	73.12	76.77	67.17	71.33
	HTSC (ours)	**57.34**	**82.11**	76.88	**75.21**	63.57	73.12	**78.13**	68.51	**71.86**
Qwen3-30B-A3B	BF16	56.23	79.04	86.88	**77.66**	64.74	79.68	80.41	**70.72**	74.42
	NVFP4-RTN	53.50	77.82	85.82	63.70	62.99	80.18	79.22	67.32	71.32
	SmoothQuant	54.18	77.10	85.52	72.85	64.04	**80.20**	78.02	68.90	72.60
	SpinQuant	53.33	77.15	73.62	65.16	**66.87**	76.88	77.75	68.75	69.94
	FlatQuant	54.18	75.21	**87.64**	64.34	64.02	79.49	78.51	69.77	71.65
	OCGQuant	55.03	78.91	86.28	71.97	64.39	79.67	78.84	69.06	73.02
	HTSC (ours)	**56.83**	**79.92**	86.96	76.88	64.12	79.91	**80.52**	70.32	**74.43**

5.3 ABLATION STUDIES

We ablate module selection, the number of compensated modules, and compensation rank on Qwen3-8B. Unless otherwise specified, all variants use the same harmful-tail teacher and training protocol. Table 4 reports the results.

Harmful-tail-guided module selection. Replacing the 14 HTSC-selected modules with randomly selected modules under a matched parameter budget reduces average downstream accuracy by 0.77 percentage points. This result supports selecting compensation sites according to their harmful-tail repair benefit rather than allocating the same capacity to arbitrary modules.

Number of compensated modules. Downstream accuracy improves as more modules are included in the HTSC selection order. We select 14 modules as the default configuration, which exceeds BF16 downstream accuracy while limiting the additional parameter and inference costs.

Compensation rank. With the 14 selected modules fixed, rank 64 achieves the best downstream accuracy among the evaluated settings, while full-rank compensation performs slightly worse despite its much larger parameter cost. We therefore use rank 64 as the default configuration.

Table 4: **Ablation studies on Qwen3-8B.** We vary module selection, the number of compensated modules, and compensation rank. Params denotes additional parameters relative to the original model (%). Avg. Acc. is the unweighted mean across eight tasks.

Study	Setting	Modules	Rank	Params (%)	Avg. Acc. (%)	WikiText-2	C4
Module selection	Random (s2)	14	64	0.13362	71.09	10.2935	15.7231
	Harmful-tail-guided	14	64	0.13362	**71.86**	10.2763	15.6989
Module count	Top-1	1	64	0.00400	70.85	10.3391	15.7911
	Top-2	2	64	0.00800	70.48	10.3427	15.7859
	Top-4	4	64	0.01840	70.90	10.3128	15.7118
	Top-8	8	64	0.05681	71.35	10.2916	15.7212
	Top-14 (default)	14	64	0.13362	**71.86**	10.2763	15.6989
Compensation rank	Rank-4	14	4	0.00835	70.73	10.3167	15.7539
	Rank-8	14	8	0.01670	70.74	10.3461	15.7762
	Rank-16	14	16	0.03341	71.30	10.2957	15.7465
	Rank-32	14	32	0.06681	71.66	10.2744	15.7038
	Rank-64 (default)	14	64	0.13362	**71.86**	10.2763	15.6989
	Full rank	14	Full	5.68408	71.58	10.2817	15.7081

6 Related Work

Existing LLM quantization methods mitigate error through reconstruction, reparameterization, or improved quantization granularity. GPTQ uses approximate second-order information for weight reconstruction (Frantar et al., 2022), while AWQ protects activation-salient weights (Lin et al., 2023). SmoothQuant balances activation and weight ranges (Xiao et al., 2023); SpinQuant learns rotations (Liu et al., 2025), and FlatQuant learns affine transformations (Sun et al., 2025) to improve quantizability. More recently, OCGQuant reduces NVFP4 blockwise error through outlier-companion channel grouping (Yao et al., 2026). Complementary approaches address error beyond individual quantization blocks: QEP studies error propagation (Arai & Ichikawa, 2025), while LQER and QERA reconstruct quantization error using low-rank corrections (Zhang et al., 2024; 2025). Unlike methods that primarily target local reconstruction error or its low-rank approximation, we study how the spectral components of accumulated final-residual error affect predictive performance. This analysis motivates HTSC, which selects compensation modules according to their harmful-tail repair benefit and distills input-conditioned corrections into sparse low-rank branches.

7 Conclusion

We show that quantization-error magnitude alone does not determine predictive degradation. Accumulated final-residual error exhibits spectral concentration and shared geometry across documents, shaped by input statistics and downstream propagation. Despite its lower energy, the spectral head is less damaging than its orthogonal tail, whose greater directional loss curvature explains the larger benefit of tail repair. Moreover, correcting only a few input-conditioned harmful-tail directions can substantially recover downstream accuracy. Motivated by these findings, we propose Harmful-Tail-Guided Sparse Compensation (HTSC), which selects modules by their propagated predictive benefit and distills selective repair into sparse low-rank branches. Across dense and mixture-of-experts Qwen3 models, HTSC restores predictive performance toward the BF16 reference under W4A4 quantization with limited additional inference cost.

AI USE STATEMENT

We used generative AI tools, including GPT-5.6 Sol and GPT-6 Astra, to assist with literature retrieval and synthesis, research framing and method design, code generation, experimental analysis and interpretation, mathematical exposition, and manuscript drafting, editing, and LaTeX preparation. The authors take responsibility for the final research methodology, experimental results, analyses, and manuscript content, including material developed with AI assistance.

REPRODUCIBILITY STATEMENT

Sections 2.1 and 2.2 describe the spectral analysis of quantization error and its formation mechanisms. Sections 3.1, 3.2, and 3.3 detail the diagnostic interventions and harmful-tail analysis. Section 4 presents the teacher construction, module selection, and compensation training procedures. Section 5.1 and Appendix A describe the quantization, calibration, and evaluation protocols.

REFERENCES

Yamato Arai and Yuma Ichikawa. Quantization error propagation: Revisiting layer-wise post-training quantization. *arXiv preprint arXiv:2504.09629*, 2025. URL https://arxiv.org/abs/2504.09629.

Saleh Ashkboos, Amirkeivan Mohtashami, Maximilian L. Croci, Bo Li, Pashmina Cameron, Martin Jaggi, Dan Alistarh, Torsten Hoefler, and James Hensman. QuaRot: Outlier-free 4-bit inference in rotated LLMs. In *Advances in Neural Information Processing Systems*, 2024. URL https://arxiv.org/abs/2404.00456.

Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013. URL https://arxiv.org/abs/1308.3432.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. PIQA: Reasoning about physical commonsense in natural language. In *AAAI Conference on Artificial Intelligence*, 2020. URL https://arxiv.org/abs/1911.11641.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try ARC, the AI2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018. URL https://arxiv.org/abs/1803.05457.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021. URL https://arxiv.org/abs/2110.14168.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. Gptq: Accurate post-training quantization for generative pre-trained transformers. *arXiv preprint arXiv:2210.17323*, 2022. URL https://arxiv.org/abs/2210.17323.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. The language model evaluation harness, 2024. URL https://zenodo.org/records/12608602.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1026–1034, 2015. URL https://openaccess.thecvf.com/content_iccv_2015/html/He_Delving_Deep_into_ICCV_2015_paper.html.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021. URL https://arxiv.org/abs/2009.03300.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. RACE: Large-scale reading comprehension dataset from examinations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2017. URL https://aclanthology.org/D17-1082/.

Jeffrey Li et al. DataComp-LM: In search of the next generation of language model training sets. *arXiv preprint arXiv:2406.11794*, 2024. URL https://arxiv.org/abs/2406.11794.

Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Wei-Ming Chen, Wei-Chen Wang, Guangxuan Xiao, Xingyu Dang, Chuang Gan, and Song Han. Awq: Activation-aware weight quantization for llm compression and acceleration. *arXiv preprint arXiv:2306.00978*, 2023. URL https://arxiv.org/abs/2306.00978.

Zechun Liu, Changsheng Zhao, Igor Fedorov, Bilge Soran, Dhruv Choudhary, Raghuraman Krishnamoorthi, Vikas Chandra, Yuandong Tian, and Tijmen Blankevoort. SpinQuant: Llm quantization with learned rotations. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2405.16406.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*, 2016. URL https://arxiv.org/abs/1609.07843.

NVIDIA. NVFP4. Transformer Engine Documentation, 2026. URL https://docs.nvidia.com/deeplearning/transformer-engine/features/low_precision_training/nvfp4/nvfp4.html.

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Ngoc Quan Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. The LAMBADA dataset: Word prediction requiring a broad discourse context. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, 2016. URL https://aclanthology.org/P16-1144/.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020. URL https://www.jmlr.org/papers/v21/20-074.html.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. WinoGrande: An adversarial winograd schema challenge at scale. In *AAAI Conference on Artificial Intelligence*, 2020. URL https://arxiv.org/abs/1907.10641.

Yuxuan Sun, Ruikang Liu, Haoli Bai, Han Bao, Kang Zhao, Yuening Li, Jiaxin Hu, Xianzhi Yu, Lu Hou, Chun Yuan, Xin Jiang, Wulong Liu, and Jun Yao. FlatQuant: Flatness matters for llm quantization. In *International Conference on Machine Learning*, 2025. URL https://proceedings.mlr.press/v267/sun25l.html.

Guangxuan Xiao, Ji Lin, Mickael Seznec, Hao Wu, Julien Demouth, and Song Han. SmoothQuant: Accurate and efficient post-training quantization for large language models. In *International Conference on Machine Learning*, 2023. URL https://arxiv.org/abs/2211.10438.

An Yang et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025. URL https://arxiv.org/abs/2505.09388.

Yishan Yao, Binjun Li, Hanling Yi, Pengyu Li, Xiaoqing Liu, Zihan Yang, Xiaotian Yu, and Zhiwen Yu. OCGQuant: Outlier-companion grouping for NVFP4 quantization. *arXiv preprint arXiv:2609.00066*, 2026. URL https://arxiv.org/abs/2609.00066.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a machine really finish your sentence? In *Annual Meeting of the Association for Computational Linguistics*, 2019. URL https://aclanthology.org/P19-1472/.

Cheng Zhang, Jianyi Cheng, George A. Constantinides, and Yiren Zhao. LQER: Low-rank quantization error reconstruction for LLMs. In *Proceedings of the 41st International Conference on Machine Learning*, 2024. URL https://proceedings.mlr.press/v235/zhang24j.html.

Cheng Zhang, Jeffrey T. H. Wong, Can Xiao, George A. Constantinides, and Yiren Zhao. QERA: an analytical framework for quantization error reconstruction. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2410.06040.

A Experimental Protocol

A.1 Quantization and Calibration

All quantized methods use W4A4 NVFP4 fake quantization with E2M1 FP4 values, a block size of 16, E4M3 block scales, and an FP32 global scale. We quantize weights and activations in 1×16 blocks for text-backbone Linear GEMMs. The `lm_head`, MoE routers/gates, embeddings, normalization layers, KV cache, and attention operations other than Linear GEMMs remain in BF16. Methods requiring calibration use 128 sequences of 2,048 tokens from WikiText-2 train, sampled with seed 0; all such methods share the same token manifest. BF16 and NVFP4-RTN require no calibration. Method-specific hyperparameters follow the original papers or official implementations, and SpinQuant uses one fixed seed.

The evaluated method matrix follows the experiment runner: Qwen3-0.6B, Qwen3-4B, Qwen3-8B, and Qwen3-30B-A3B (Yang et al., 2025) include all six baselines. HTSC is included for all four models.

A.2 Downstream Evaluation

ARC-C and ARC-E (Clark et al., 2018), HellaSwag (Zellers et al., 2019), PIQA (Bisk et al., 2020), and WinoGrande (Sakaguchi et al., 2020) use canonical zero-shot likelihood-based evaluation. LAMBADA (Paperno et al., 2016) uses `lambada_openai` at zero shot. MMLU (Hendrycks et al., 2021) uses canonical five-shot multiple-choice evaluation. GSM8K (Cobbe et al., 2021) uses five-shot greedy generation with the Qwen chat template, `enable_thinking=False`, `fewshot_as_multiturn=True`, and no additional system prompt. The diagnostic experiments in Section 3.1 additionally use RACE (Lai et al., 2017).

A.3 Language-Modeling Evaluation

WikiText-2 test is evaluated in non-overlapping 2,048-token chunks, discarding the final shorter chunk. C4 validation uses a fixed draw of 256 samples of 2,048 tokens with seed 0. Both perplexities are computed from the total negative log-likelihood over all valid target tokens with batch size 1.

B HTSC Compensation Training Details

This section specifies the training settings accompanying Section 4. Quantization and downstream evaluation protocols are described separately in Appendix A.

Training data and held-out split. Calibration-based methods and HTSC use the same 128 sequences of 2,048 tokens sampled from the WikiText-2 training split with seed 0. HTSC additionally uses 512 held-out sequences, each containing at most 512 tokens, drawn from the WikiText-2 training split and disjoint from both the shared training-token set and the WikiText-2 test split used for reporting perplexity. Held-out sequences are excluded from basis estimation, module screening, finite-correction verification, and gradient updates.

Sequences containing fewer than two tokens are excluded. Each sequence supplies observed next tokens as causal targets, and padding positions are excluded from all losses. Sequence identifiers, data partitions, and cross-fitting folds are fixed before training. Sequences from the same source document are kept together when assigning partitions and cross-fitting folds.

Teacher preparation. Teacher construction uses four-fold document-level cross-fitting within the training partition. The structured-head rank is 32, and at most 32 tail directions with strictly positive predicted benefit are selected per document. For a residual dimension d, the complementary tail has dimension $d - 32$. Documents with fewer than 32 positive-benefit directions use only the available positive-benefit directions.

The BF16 reference and uncompensated quantized model are frozen. Paired residuals, cross-fitted bases, and document-specific repair targets are prepared before branch optimization and remain fixed. Module selection and distillation use the same teacher targets and harmful-tail projectors; no separate selection teacher is fitted.

Module-selection settings. Candidate sites are attention and MLP linear projections, excluding embeddings and the output language-model head. We use eight training sequences for gradient screening and a disjoint set of 16 training sequences for finite-correction verification. Both subsets belong to the shared 128-sequence training set rather than the held-out checkpoint-selection partition.

The 28 highest-scoring candidates from gradient screening are verified using a finite probe magnitude of $\epsilon = 0.1$. Candidates with positive mean verified benefit are ordered by verified benefit divided by their original dense weight parameter count. We greedily retain candidates subject to a maximum of 14 modules and a cumulative dense-weight budget of $1/12$ of the total candidate linear-weight parameters. This budget controls the selected sites rather than directly constraining the additional low-rank parameter count. The selected module set is fixed throughout compensation training.

Branch initialization and precision. The default compensation rank is 64. Each input factor A_m uses Kaiming-uniform initialization (He et al., 2015) with $a = \sqrt{5}$, and each output factor B_m is initialized to zero. The branch multiplier is one, without additional rank-dependent scaling. Consequently, the initial compensated model reproduces the uncompensated quantized model.

Branch master parameters are stored in FP32, while branch operands and intermediate outputs are rounded to BF16 during forward computation. The backbone, normalization layers, and final readout remain frozen. Dropout is disabled. Gradients propagate through frozen quantized linear layers using the activation straight-through estimator (Bengio et al., 2013), without updating the quantized backbone weights. Only compensation parameters are passed to the optimizer.

Optimization settings. We use AdamW (Loshchilov & Hutter, 2019) with a peak learning rate of 10^{-5}, $(\beta_1, \beta_2) = (0.9, 0.999)$, optimizer epsilon 10^{-8}, and zero weight decay. Each optimizer step processes one 2,048-token sequence, with no gradient accumulation. Two epochs over the 128 training sequences therefore produce 256 optimizer steps. The second epoch deterministically reshuffles the training list. The base random seed is 20260923; the shuffle seed is the base seed plus the zero-based epoch index. The global gradient norm of trainable parameters is clipped to 1.0 before each update.

The learning rate increases linearly during the first 13 steps, then follows cosine decay to 10^{-6}. For update $s \in \{1, \ldots, S\}$, with $S = 256$ and $w = 13$, the schedule is

$$\eta_s = 10^{-5} \begin{cases} s/w, & s \le w, \\ 0.1 + 0.45 \left[1 + \cos\left(\pi \frac{s-w}{S-w}\right)\right], & s > w. \end{cases} \tag{7}$$

The distillation temperature is one. We optimize full-vocabulary teacher-to-student KL divergence, averaged over valid prediction positions within each sequence. No additional ranking, multiple-choice, or supervised cross-entropy loss is added to the distillation objective. Observed next-token labels are used for teacher construction and held-out NLL computation. Readout calculations are processed in chunks of 32 token positions to reduce temporary memory consumption; all valid positions and the full vocabulary remain included.

Checkpoint selection. Held-out next-token NLL is evaluated at steps $0, 16, 32, 64, 128, 192$, and 256. For WikiText-2, this score is computed by summing token negative log-likelihoods across all 512 held-out sequences and dividing by the total number of valid prediction tokens.

We select the checkpoint with the lowest held-out NLL. An exact tie retains the earlier checkpoint, and the uncompensated step-zero model is eligible for selection. Training runs for the complete schedule; the selected checkpoint is not necessarily the final checkpoint. Neither downstream task accuracy nor test-set perplexity is used for checkpoint selection.

At validation milestones, continuation states save branch parameters, optimizer states, the update index, and the best-checkpoint record. The deterministic schedule reconstructs the learning rate and data order when training resumes.

Ablation controls. Ablations share the same data partitions, teacher targets, optimization schedule, and checkpoint-selection criterion. Module-count ablations use prefixes of the verified module ordering. Rank ablations retain the selected sites and vary only the compensation rank. The full-rank variant uses one zero-initialized dense additive matrix per selected module.

Random-selection runs sample the same number of candidate modules without replacement. Random selection changes module selection rather than the shared training-data partition. Because candidate modules have different dimensions, equal module counts do not imply equal additional parameter counts.

C Downstream Results on Smaller Models

Table 5 reports downstream accuracy on Qwen3-0.6B and Qwen3-4B, complementing the results on larger models in Table 3. All quantized methods follow the W4A4 NVFP4 protocol described in Appendix A.

Table 5: **Downstream accuracy (%) on smaller Qwen3 models.** Results on Qwen3-0.6B and Qwen3-4B under W4A4 NVFP4 quantization. Avg. is the unweighted mean across eight tasks, except for Qwen3-0.6B, whose average is computed across the seven reported tasks. Higher is better.

Model	Method	ARC-C	ARC-E	GSM8K	HellaSwag	LAMBADA	MMLU	PIQA	WinoG.	Avg.
Qwen3-0.6B†	BF16	**34.13**	56.14	-	47.36	**40.03**	**47.29**	**67.52**	57.06	**49.93**
	NVFP4-RTN	29.27	49.75	-	43.53	27.46	42.52	61.26	53.67	43.92
	SmoothQuant	29.10	49.96	-	43.77	30.06	41.56	62.13	53.83	44.34
	SpinQuant	29.95	51.26	-	39.99	28.08	35.44	60.99	53.91	42.80
	FlatQuant	30.12	45.12	-	43.65	33.40	42.81	63.82	53.83	44.68
	OCGQuant	30.63	43.39	-	44.18	31.01	41.03	62.51	52.49	43.61
	HTSC (ours)	33.96	**57.03**	-	**47.62**	35.84	46.11	66.97	**57.54**	49.30
Qwen3-4B	BF16	**54.10**	78.41	59.44	**68.36**	59.48	**70.16**	74.97	**65.82**	66.34
	NVFP4-RTN	50.17	74.33	39.73	65.73	53.02	66.94	73.29	61.88	60.64
	SmoothQuant	49.06	72.31	49.51	65.40	54.76	66.76	72.03	60.85	61.33
	SpinQuant	44.54	64.52	46.25	65.50	59.11	63.84	72.36	63.77	59.99
	FlatQuant	49.32	72.56	39.58	65.84	56.26	67.33	72.31	63.22	60.80
	OCGQuant	49.23	74.20	**69.90**	65.84	56.86	67.07	71.38	61.48	64.50
	HTSC (ours)	53.92	**79.29**	61.18	68.05	**59.83**	69.44	**75.57**	64.96	**66.53**

† GSM8K results are omitted for Qwen3-0.6B because the model is too small to produce meaningfully discriminative performance on this benchmark. The reported Avg. for Qwen3-0.6B is therefore computed over the remaining seven tasks.

D Language-Modeling Perplexity Results

We evaluate language-modeling perplexity on WikiText-2 and C4 across all four Qwen3 models. Table 6 compares HTSC with the BF16 reference and the uncompensated NVFP4-RTN baseline, complementing the downstream accuracy results reported in Section 5.2.

Table 6: **Language-modeling perplexity.** WikiText-2 and C4 perplexity across Qwen3 models. Lower is better.

	WikiText-2 ↓			C4 ↓		
Model	BF16	NVFP4-RTN	HTSC	BF16	NVFP4-RTN	HTSC
Qwen3-0.6B	21.033	28.031	22.094	27.849	35.718	29.384
Qwen3-4B	13.668	14.252	14.143	18.170	18.971	18.897
Qwen3-8B	9.940	10.367	10.276	15.102	15.651	15.699
Qwen3-30B-A3B	8.729	9.190	9.041	13.289	13.801	13.782